Next Best View for Apple Estimation

ROB545 Final Project

Keegan Nave
MIME Department
Oregon State University

Luke Strohbehn
MIME Department
Oregon State University

Pico Sankari
MIME Department
Oregon State University

I. Project Summary

A. Introduction

Agricultural robotics have become a key topic for industry and academia due to labor shortages, rising costs, and supply chain bottlenecks. Apple orchards have been hit especially hard by labor shortages during both pruning and picking season which makes them a perfect target for robotic automation. However, orchards are complicated environments with varying structure, lighting, and crop conditions which can make perception and manipulation tasks difficult [1], [2]. To increase effectiveness and to prevent damage to equipment and crops, it is necessary to have a proper representation of the scene to make informed decisions.

Although scene understanding approaches using computer vision and deep learning have improved in recent years [3], there are often still gaps left in scene representations due to sensor noise and environmental variance [4]. Because this problem persists across approaches and sensor types [3], active scanning has become a popular approach to reduce uncertainty while imaging. One of the most popular approaches for active scanning is Next Best View [2], [4], [5], which predicts the next best imaging location using an uncertainty reduction heuristic [6] and previous scans of the environment.

B. Problem Statement

For this project we implemented a Next Best View Algorithm using a UR5e and RGB-D camera in a simulation environment to scan points on apples and find their volumetric estimations.

This approach could be used to improve fruit yield estimation, and the base algorithm can be modified for future research/projects to work with branch scanning.

C. Prior Work

NBV algorithms have been the subject of many papers and research projects over the past few decades. There have been many approaches with a variety of hardware in a broad range of environments. One approach that was very similar to our project used RGB-D data and a UR5 arm in an orchard environment to determine the size of apple fruitlets (small unripe apples) [7]. This research used a stereo camera mounted on the end of the UR5 arm to scan fruit clusters. The clusters had been previously marked with AprilTags to allow the robot to locate them. The NBV algorithm was used to collect scans of the clusters. The data was then processed into point clouds that were segmented using Mask R-CNN. The segmented point clouds were then translated to estimated volumes. This technique proved accurate, generating R^2 values of around 0.9 when comparing the estimated and ground truth values.

Menon et al. used RGB-D data and a UR5 arm to estimate the size of peppers [8]. This paper explored the environment without markers on the fruits and attempted to find fruits using Mask R-CNN. Once detected, the fruits were mapped to estimated shapes during the NBV scans. The results were mediocre, resulting in volumetric accuracy of around 60% across three different scenarios. Further efforts to reconstruct plants and their fruits have been explored [1], [9]–[11], but each must balance the trade-off between reconstruction accuracy and computational complexity.

D. Algorithm Description

We completed this work in a Gazebo simulation environment, as seen in Figure 1. The environment contained a

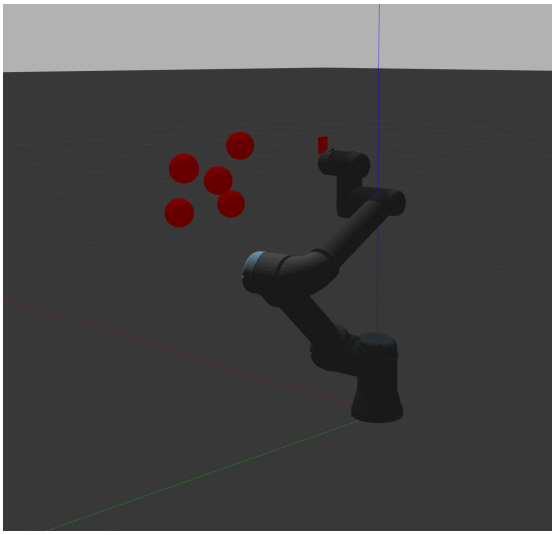

Fig. 1. UR5e with camera camera attached to the end-effector (red rectangular prism) in the simulation's initial view pose. Apples are represented as red spheres.

ROB 545, Spring 2023. All code can be found at the following repository: https://github.com/Keeganfn/apple-nbv

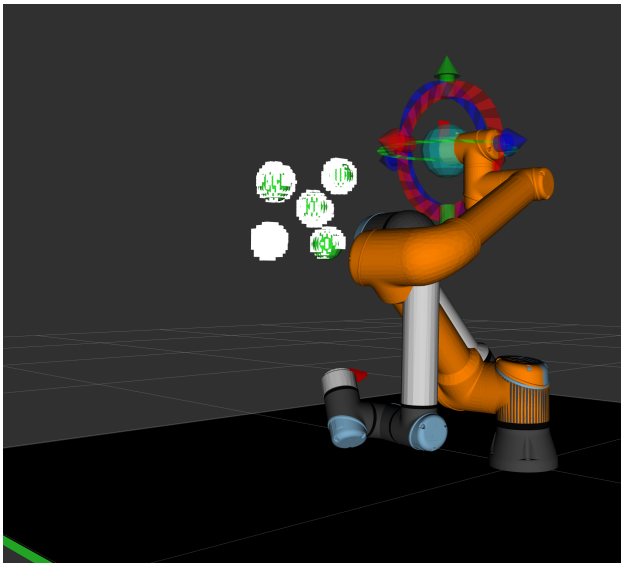

Fig. 2. UR5e in RViz after running the next best view algorithm. Green obstacles are added to the scene to make sure the next path does not interfere with an apple.

simulated UR5e arm with a RGB-D camera at the end. As the arm moved, the scans from the camera were used to update the simulation's Octomap. An Octomap is a 3-D occupancy map that tracks which voxels (3 dimensional grid spaces) are occupied in space. An example is demonstrated in Figure 2 with the white cubes in RViz representing an occupied voxel. Simulated apples were modeled as red spheres and placed in our simulation environment. Our algorithm's goal was to plan the optimal next viewing location. This was accomplished in several steps:

- The environment is initialized and the camera takes the first scan. All scans are filtered to only select red (255,0,0) pixels, removing background noise and returning only pixels that belong to the apples. This scan creates the Octomap.
- The pixels in the Octomap are clustered using a RANSAC algorithm. Each cluster represents an apple and contains points in [X, Y, Z] form.
- Each cluster is fit to a sphere. The returned sphere has an XYZ center and a radius. This is the estimate of the apple based on the current octomap.
- Each sphere is divided into an upper and lower hemisphere. These hemispheres are then divided into 8 equal sections, resulting in 16 total sections. The points in each cluster are then binned into their sphere's sections, resulting in a count of points in each bin.
- The bin with the least points (when considering all front facing bins for all spheres) is then selected. Only front facing bins are considered because it is not possible for the UR5e to reach behind the apples. Additionally, a list of all visited bins is kept and any bin already visited is ineligible to be visited again.
- A unit vector that points through the center of the selected bin is generated. This unit vector is the vector that the camera will attempt to orient to in order to view the bin.
- This vector is scaled by the distance from the center of the sphere being viewed to the base frame of the robot. This distance was selected to keep the robot rotating around the fruits at a distance that was both close to reachable space and allowed for observation of multiple fruits at once. The scaled vector is added to the coordinates of the center of the observed sphere, resulting in the coordinates that the robot end-effector will try to reach.
- The resulting points are filtered to return reachable coordinates. Any points that are below 0.2m in the Z direction are set to have a Z value of 0.2m to prevent the robot from running into the ground. Next, the distance from the base of the robot to the desired coordinates is checked. If this distance is further than the arm can reach, the coordinates are scaled down to a reachable position. The camera orientation between this reachable position and the desired sphere is then established and used as the desired camera orientation.
- The arm moves to the desired coordinates and orientation. The Octomap is continually updated as the arm moves.
- After a movement is complete, the bin selection algorithm is repeated and the arm moves to the resulting coordinates and orientation. Once every bin is full (has at least 100 points) or has been visited, the algorithm completes.
- Resulting volumetric estimates for the apples in the point cloud, as well as their locations in space are returned.

II. Results

Although we knew exact size and location of the red spheres in the Gazebo environment, it is still impossible for the end-effector-mounted camera to view every square inch of each sphere's surface area, and point cloud detection may be unable to fill every voxel defining the surfaces of the spheres. Therefore, we developed a "optimal information gain" baseline trial by dragging around the end-effector in RViz and executing trajectories for what we defined as a human next best view. The number of poses was unlimited, and data was captured until we surmised there were no more viewable and unoccupied voxels. This created a control scan that was deemed to be the best results a hypothetical scan could return. The control-based size values of the spheres do in fact fall below the actual sizes of the spheres in Gazebo, likely due to the volumetric approximation that accompanies the use of octomaps.

Next, we ran two sets of control trials – one performed by taking two additional views after the starting pose, the other with three additional views – each of which navigated the end-effector to a random pose with the yz-plane located at the initialization point of the end-effector. No rotations were applied to the end-effector during these random movements. It should be noted that these "random" movements were randomly selected by the user, and not randomly generated. A more in depth study should execute more random trials for a better average. These trials are referred to as 'Random 2 Moves' and 'Random 3 Moves' in the figures.

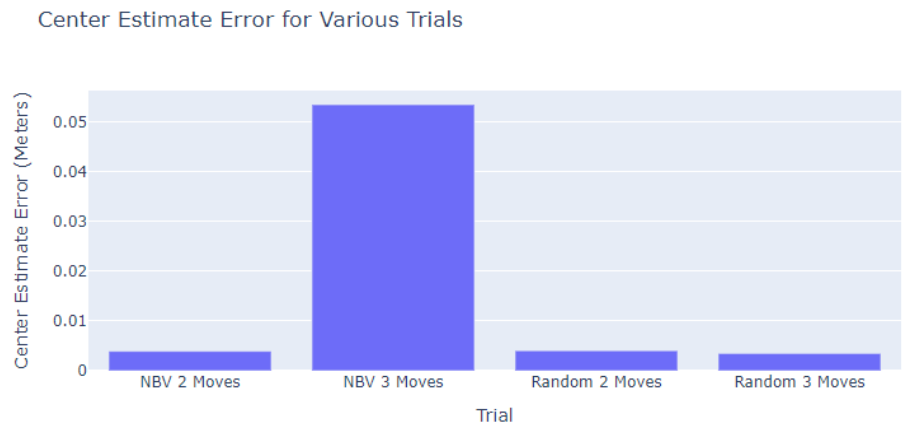

Fig. 3. Sphere center estimation (outlier included)

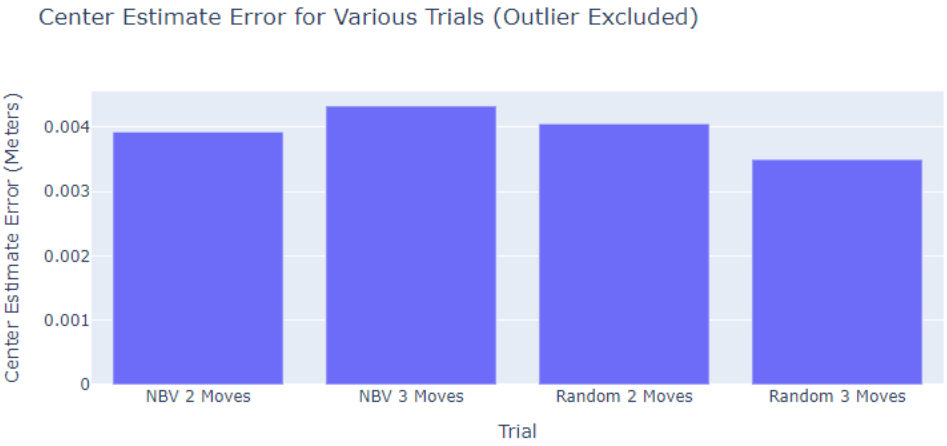

Fig. 4. Sphere center estimation (outlier excluded)

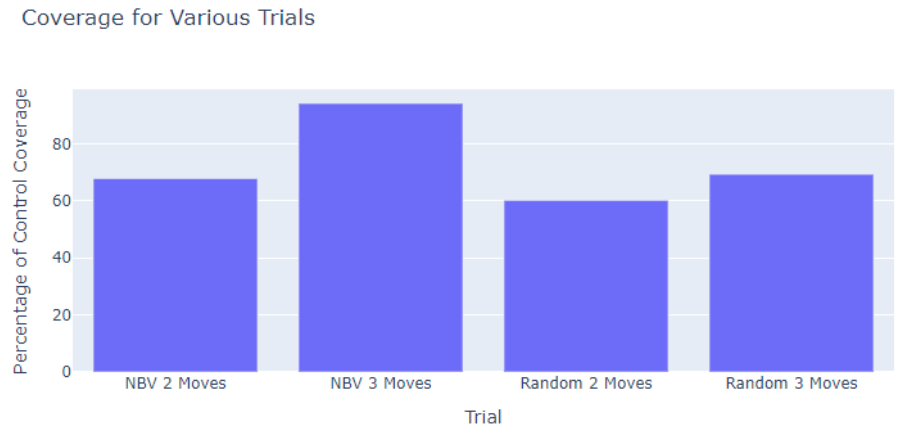

Fig. 5. Point cloud coverage by sphere (outlier excluded)

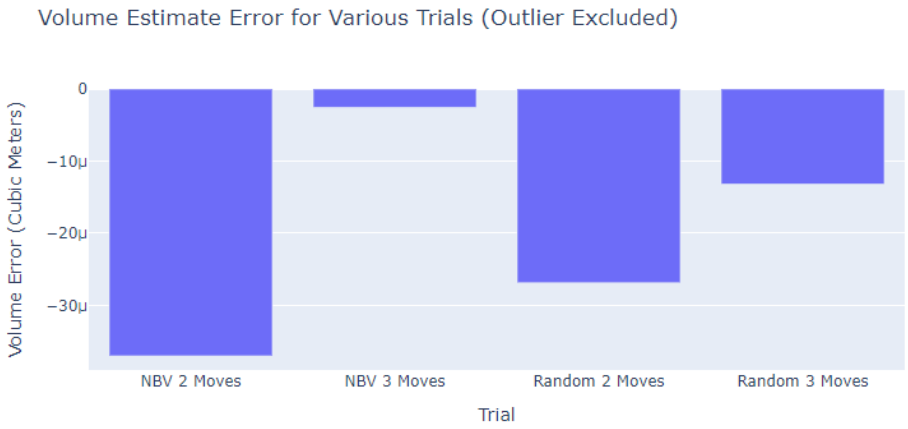

Fig. 6. Sphere volume estimate error (outlier excluded)

The next best view trials were run using the exact same parameters as the control and baseline trials, but selected the orientations and locations of the end effector using our algorithm. These trials are referred to as 'NBV 2 Moves' and 'NBV 3 Moves' in the figures. During one of the three-view trials, the RANSAC algorithm used for clustering did not detect one of the spheres, and another detected sphere subsequently counted the number of points, wildly throwing off our results, as seen in Figure 3. Removal of this outlier gave us an improved result, as seen in Figure 4, demonstrating that the three-view NBV reduced the total amount of error. Similarly, the volumetric estimation error (Figure 6) demonstrates the improved object identification with an additional view. However, more trials would need to be run in order to validate these initial results. These next best view trials resulted in substantially more exploration of the spheres when compared to random movements, as seen in Figure 5. The coverage values in this figure were computed by dividing the number of points on the each sphere returned by the trial movements by the points on each sphere returned by our fully explored control trial.

III. Conclusion/Future Work

A. Discussion

This research was confined to an idealized simulation environment. This provided a base for getting volumetric estimates of apples, but would need further work to be deployed in the real world. The first step in this work would be adding occlusions into the simulated environment. In the real world, apples will almost always be partially obscured by branches and/or leaves. Simulated branches and leaves could be added to the environment to test the effect of occlusions on the current algorithm. Based on results, the algorithm may need to be modified. This research also used an idealized environment to make the segmentation of apple points from non-apple points very easy. To reach deployment in the real world, a more sophisticated segmentation approach would be needed. This would most likely be in the form of a neural network (such as YOLO) trained to segment apples from their background. In the real world, clustering of the point clouds into apples may become more difficult. In the case that an apple was split in two by an occlusion, the clustering algorithm would need to be updated to attempt to group these two separate clusterings into one. Another potential issue regards the state space used in the simulation environment. In the simulation, all of the apples were within view of the initial scan. In reality, this would not be the case. This could be partially rectified by beginning the scan process with a few poses that are on the edge of the reachable space and would maximize the area scanned and the number of apples detected. If an even larger space needs to be explored, the incorporation of a moving platform that the UR5e is attached to may be necessary. Despite a number of adaptations needed for real world deployment, this algorithm provides a strong framework for future research and deals with the technical work required to make this system work in ROS, which is not trivial.

B. Conclusion

This research resulted in an effective algorithm for selecting a next best view for creating volumetric estimates of apples. The algorithm runs quickly, taking an average of less than 1 second to select the next position. The algorithm explored the space more efficiently than random movements, yielding 94% of all points available after 3 moves as opposed to 3 random moves returning 69% of available points. In the future, this work could be extended into real orchards and help advance precision agriculture efforts.

References

[1] F. P. Boogaard, K. S. A. H. Rongen, and G. W. Kootstra, "Robust node detection and tracking in fruit-vegetable crops using deep learning and multi-view imaging," *Biosystems Engineering*, vol. 192, pp. 117–132, Apr. 2020.

[2] J. Hemming, J. Ruizendaal, J. W. Hofstee, and E. J. van Henten, "Fruit detectability analysis for different camera positions in sweet-pepper," *Sensors (Basel, Switzerland)*, vol. 14, no. 4, pp. 6032–6044, Mar. 2014.

[3] S. Aarthi and S. Chitrakala, "Scene understanding — a survey," in *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 2017, pp. 1–4.

[4] M. Hržica, R. Cupec, and I. Petrović, "Active vision for 3D indoor scene reconstruction using a 3D camera on a pan-tilt mechanism," *Advanced Robotics*, vol. 35, no. 3-4, pp. 153–167, Feb. 2021.

[5] M. Karaszewski, M. Adamczyk, and R. Sitnik, "Assessment of next-best-view algorithms performance with various 3d scanners and manipulator," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 119, pp. 320–333, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271616301307

[6] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1016–1030, Oct. 1999.

[7] H. Freeman and G. Kantor, "Autonomous apple fruitlet sizing with next best view planning," 2023.

[8] R. Menon, T. Zaenker, N. Dengler, and M. Bennewitz, "Nbv-sc: Next best view planning based on shape completion for fruit mapping and reconstruction," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 4197–4203.

[9] D. Gu, K. Zhu, Y. Shao, W. Wu, L. Gong, and C. Liu, "3D Scanning and Multiple Point Cloud Registration with Active View Complementation for Panoramically Imaging Large-Scale Plants," in *Intelligent Robotics and Applications*, ser. Lecture Notes in Computer Science, H. Yu, J. Liu, L. Liu, Z. Ju, Y. Liu, and D. Zhou, Eds. Cham: Springer International Publishing, 2019, pp. 329–341.

[10] A. K. Burusa, E. J. van Henten, and G. Kootstra, "Attention-driven Active Vision for Efficient Reconstruction of Plants and Targeted Plant Parts," Jun. 2022.

[11] S. Foix, G. Alenyà, and C. Torras, "Towards plant monitoring through Next Best View," *Frontiers in Artificial Intelligence and Applications*, vol. 232, Jan. 2011.